6. 5,438,654, Aug. 1, 1995, System and method for sharpening texture imagery in computer generated interactive graphics; Robert A. Drebin, et al., 395/139, 128, 133 [IMAGE AVAILABLE]

8. 5,179,638, Jan. 12, 1993, Method and apparatus for generating a **texture** **mapped** perspective view; John F. Dawson, et al., 395/125, 127, 130 [IMAGE AVAILABLE]

10. 4,727,365, Feb. 23, 1988, Advanced video object generator; William M. Bunker, et al., 345/139 [IMAGE AVAILABLE]

```
=> d his
     (FILE 'USPAT' ENTERED AT 16:07:07 ON 06 MAY 1997)
L1          197 S TEXTURE (4A) MAP####
L2          258 S (BILINEAR OR BI LINEAR) (4A) INTERPOLAT###
L3           19 S L1 AND L2
L4           11 S L3 AND (LUT OR LOOK UP)
=>
```

updated 3/5/02

L2 1 L1 AND TEXTURE#
=> d kwic
US PAT NO: **5,204,944** [IMAGE AVAILABLE] L2: 1 of 1

SUMMARY:

BSUM(4)

Image . . . include distortion compensation of imaging sensors, decalibration for image registration, geometrical normalization for image analysis and display, map projection, and **texture** mapping for image synthesis.

SUMMARY:

BSUM(20)

There . . . single input value is correctly handled when undergoing magnification. For a review of these techniques, see Heckbert, P., "Survey of **Texture** Mapping", IEEE Computer Graphics and Applications, vol. 6, no. 11, pp. 56-67, November 1986.

SUMMARY:

BSUM(61)

The . . . (b) maximized the area of the intermediate image, it actually caused more severe aliasing (see Smith, A. R., "Planar 2-Pass **Texture** Mapping and Warping," Computer Graphics, (SIGGRAPH '87 Proceedings), vol. 21, no. 4, pp. 263-272, July 1987). This non-intuitive result is. . .

DETDESC:

DETD(9)

In . . . for the user to specify ZLUT which associates a z coordinate value with each pixel. This allows warping of planar **textures** onto non-planar surfaces, and is useful in dealing with foldovers. The objective, however, is not to solve the general 3-D. . .
=> d

1. **5,204,944**, Apr. 20, 1993, Separable image warping methods and systems using spatial lookup tables; George Wolberg, et al., 395/127; 345/139; 382/276, 277, 304; 395/119 [IMAGE AVAILABLE]
=> d his

(FILE 'USPAT' ENTERED AT 08:58:32 ON 21 MAR 1997)

L1 1 S 5204944/PN

L2 1 S L1 AND TEXTURE#

=> s l1 and fraction##

197779 FRACTION##

L3 1 L1 AND FRACTION##

=> d kwic

US PAT NO: **5,204,944** [IMAGE AVAILABLE] L3: 1 of 1

DETDESC:

DETD(27)

If . . . fit in one output pixel, compute the location of the next output pixel boundary and update the current position and **fraction** of input pixel remaining. Using the gradient information compute the value of the interpolated intensity half way to said boundary,. . .

DETDESC:

DETD(34)

As . . . 8(a), at 200, it is assumed that computations are done on n bit fixed point numbers with f bits of **fraction**, d bits of integer, and one sign bit. It is further assumed that the number is in signed magnitude representation,. . .

DETDESC:

DETD(42)

When enabled, the trunc units 208,211, set the **fractional** parts of their respective fixed point inputs to zero, see FIG. 8(d) 203. When disabled they pass the data through.. . .

DETDESC:

DETD(46)

Unit 226 is an f bit AND gate that outputs a single bit signifying whether or not the **fractional** part of curL is zero. It is connected to the last f bits of the connection from trunc unit 208. . .

DETDESC:

DETD(60)

The multiplier unit with storage 218 multiplies the f bit **fraction**

on its left input (connected to the **fractional** field of dist 220) by the n bit fixed point number on its right input (from the storage of division. . .

DETDESC:

DETD(94)

A . . . bottleneck flag is set to one, then that pixel makes no contribution to B.sub.x. The bottleneck image thus reflects the **fraction** of each pixel in the intermediate image not subject to bottleneck distortion in the first pass. The computations are straightforward,. . .

DETDESC:

DETD(102)

In . . . of potential folds may be large. Furthermore, it is often the case that the folded area may represent a small **fraction** of the output image. Thus, using one frame buffer per fold would be both inefficient and very expensive.

```
=> s l1 and (fraction## (p) lut)
        197779 FRACTION##
          1216 LUT
            26 FRACTION## (P) LUT
L4           0 L1 AND (FRACTION## (P) LUT)
=> s l1 and lut
          1216 LUT
L5           0 L1 AND LUT
=> s l1 and (fraction## (p) (xlut or ylut))
        197779 FRACTION##
             8 XLUT
             8 YLUT
             0 FRACTION## (P) (XLUT OR YLUT)
L6           0 L1 AND (FRACTION## (P) (XLUT OR YLUT))
=> s l1 and (xlut or ylut)
             8 XLUT
             8 YLUT
L7           1 L1 AND (XLUT OR YLUT)
=> d kwic
```

US PAT NO: **5,204,944** [IMAGE AVAILABLE] L7: 1 of 1

DETDESC:

DETD(4)

The . . . resampler 28 warps or transforms y coordinate data, which will be described in the context of a spatial lookup table **YLUT**, so that the second pass via intensity sampler 32 can access coordinate data without the need for an inverse function.. . .

DETDESC:

DETD(7)

In . . . are shown as input terminals 11, 12 and 13 for respectively supplying data for input image I, x value data **XLUT**, and y value data **YLUT**. Generally, the input image data may represent values of intensity or luminance of a pixel or a plurality of pixels. . . a three color component system such as red, green and blue, or any values relating to an image. Similarly, the **XLUT** input data may represent values of a first coordinate to which pixels of the input image are to be transferred in an output image and the **YLUT** data may represent such values for a second coordinate.

DETDESC:

DETD(8)

Scanline . . . and Y. Sampling X and Y over all input points yields data representative of two new real-valued images, designated as "**XLUT**" and "**YLUT**", specifying the point-to-point mapping from each pixel in the input image onto the output image. **XLUT** and **YLUT** will be referred to as spatial lookup tables since they can be viewed as 2-D tables which express a spatial. . .

DETDESC:

DETD(9)

In addition to **XLUT** and **YLUT**, as will be further discussed with reference to FIG. 3, provision is also made for the user to specify ZLUT. . .

DETDESC:

DETD(11)

For each pixel (u, v) in input image I, spatial lookup tables **XLUT**, **YLUT**, and ZLUT are indexed at location (u, v) to determine the corresponding (x, y, z) position of the input point. . . The z coordinate will only be used to resolve foldovers. This straightforward indexing applies only if the dimensions of I, **XLUT**, **YLUT** and ZLUT

are all identical. If this is not the case, then the smaller images are upsampled (magnified or expanded). . .

DETDESC:

DETD(20)

As an example, consider the 1-D arrays shown in FIG. 6. The first row is taken from **XLUT**, the second from **YLUT**, and the third from input intensity image I. The next two arrays show **YLUT** and I resampled according to **XLUT**.

DETDESC:

DETD(67)

In FIG. 2, input image I is shown being warped according to **XLUT** to generate intermediate image I.sub.x. In order to apply the second pass, **YLUT** is warped alongside I.sub.x, yielding **YLUT**.sub.x. This resampled spatial lookup table is applied to I, in the second pass. The result is output image I.sub.xy. Referring. . . intermediate pixel value data representing the input image pixel values of image I after being warped or resampled according to **XLUT** to give effect to image compression variations along the direction of a first coordinate x, to generate intermediate image I.sub.x,. . . pixel value data representing the intermediate image pixel values from output 27 of means 26 after being resampled according to **YLUT**.sub.x, to give effect to image compression variations along the direction of the second coordinate y, to generate preliminary output image. . .

DETDESC:

DETD(70)

YLUT.sub.x is computed in the coordinate resampler means 28 depicted in the second row of channel 20 in FIG. 2. The ability to resample **YLUT** for use in the second pass has important consequences: it circumvents the need for a closed-form inverse of the first. . . the u coordinate associated with a pixel in the intermediate image. Thus, instead of computing the inverse to index into **YLUT**, we simply warp **YLUT** into **YLUT**.sub.x, allowing direct access in the second pass.

DETDESC:

DETD(71)

The . . . similar to the intensity resampler 26. It differs only in

the notable absence of antialiasing filtering--the output coordinate values in **YLUT**.sub.x are computed by point sampling **YLUT**. Interpolation is used to compute values when no input data is supplied at the resampling locations. However, unlike the intensity. . . coordinate values of other contributions to that pixel. This serves to secure the accuracy of edge coordinates as represented in **YLUT**.sub.x, even when the edge occupies only a partial output pixel.

DETDESC:

DETD(72)

The following example demonstrates the coordinate resampling algorithm. Consider the arrays shown before in FIG. 6. **YLUT**.sub.x in the example is the output of the coordinate resampling as computed below. Notice that the output consists of point. . .

DETDESC:

DETD(73)

As . . . of FIG. 3, we also apply this resampling to ZLUT in exactly the same manner as it was applied to **YLUT**, as will be discussed further.

DETDESC:

DETD(81)

This . . . computed from the sparse samples by interpolation. For example, linear interpolation can be used to increase the spatial resolution of **XLUT** and **YLUT**. The resulting image in FIG. 13 is shown to be antialiased, and clearly superior to its counterpart in FIG. 12.. . .

DETDESC:

DETD(88)

If . . . due to horizontal shearing and/or perspective need to be considered in this case. The vertical scale factor, vfctr , for **XLUT** and **YLUT** is given by vfctr=MAX(.DELTA.X.sub.AC, .DELTA.X.sub.BD). Briefly this measures the maximum deviation in the horizontal direction for a unit step in. . .

DETDESC:

DETD(98)

Up . . . For instance, consider the 1-D arrays shown in the following table. They denote the input intensities I and their respective **XLUT** coordinate values for a given image row.

DETDESC:

DETD(99)

XLUT=0.6, 2.3, 3.2, 2.0, 4.2

DETDESC:

DETD(101)

Unlike the example in FIG. 6, note that **XLUT** is not monotonic. that is, **XLUT** now specifies a 1-D path which folds back upon itself, as the **XLUT** values successively increase, then decrease, then increase. In particular, the first three input pixel values are resampled and stored in. . . left-to-right order from x=2.0 to x=4.2. Thus, two foldovers are present because two sign changes exist between adjacent entries in **XLUT**. Of course, when **XLUT** is monotonically increasing or decreasing, no foldovers exist.

DETDESC:

DETD(103)

To . . . are to be displayed. The simplest mechanism, and probably the most useful, is to make provision for supplying not only **XLUT** and **YLUT**, but also ZLUT in order to specify the output z coordinate for each input pixel. In the first pass ZLUT will be processed in the same manner as **YLUT**, so the second pass of the intensity resampler can have access to the z-coordinates. Thus, as shown in FIG. 3,. . .

DETDESC:

DETD(106)

The . . . front layer, which we shall refer to as the zero foldover layer, is initalized with the first left-to-right span between **XLUT** values x=0.6 and x=3.2. the next span, from x=3.2 to x=2.0, forces several columns to become multi-valued as they are. . .

DETDESC:

DETD(111)

In order to signify the end of a vertical span, the negative value of the y-coordinates of **YLUT**.sub.x are stored. Since negative y-coordinates are considered invalid in this system (offsets must be added to make negative coordinates become. . .

DETDESC:

DETD(112)

This . . . the data structure for vertical scanlines. Notice that the multiple channels of information that comprise the intermediate image (I.sub.x, B.sub.x, **YLUT**.sub.x, ZLUT.sub.x) are collapsed into one structure for convenience. It is important to note that image I.sub.x can actually contain color. . .

DETDESC:

DETD(133)

Input . . . pixel of the image is to be transferred in the output image. In FIG. 2, these values are denoted as **XLUT** and **YLUT** which are characterized as look-up tables of the values of the first and second coordinates, x and y. These coordinates. . .

DETDESC:

DETD(135)

Shear . . . in the image for developing at first and second outputs 23 and 24 scaled x and y coordinate data representing **XLUT** and **YLUT** values of a magnified output image. By providing a plurality of pixel values in place of each pixel value of the basic output image, as by interpolation of input coordinate samples as discussed above, the spatial resolution of **XLUT** and **YLUT** is, in effect, increased to alleviate jagged edges indicative of undersampled coordinate data. Shear resampler 22 is also coupled to. . .

DETDESC:

DETD(136)

"Basic . . . output image in its desired final format, (i.e.--before any magnification of the final image). First coordinate data is identified as **XLUT** at input terminal 12, as well as at output 23 of shear resampler 22. Although the data at output 23. . . magnification,

depending on the presence of shear conditions, for simplicity the same label is used. The same is true for **YLUT** (and ZLUT in FIG. 3).

DETDESC:

DETD(138)

Coordinate . . . shear resampler 22 and functions to develop at a first output 29 modified second coordinate data. This data, indicated as **YLUT**.sub.x, represents the scaled **YLUT** values from resampler 27 after resampling in resampler 28 responsive to **XLUT** values to give effect to image compression variations occurring along the direction of the first or x coordinate. At its second output 30 the coordinate resampler 28 develops a signal B.sub.x, which represents excised y coordinate data representing **YLUT** data from putput 24 which has been resampled so as to delete values for pixels subject to positive compression variations. . .

DETDESC:

DETD(141)

The . . . 15 transposed input image value data, indicated as I.sup.T. The transposed first and second coordinate data are similarly developed as **XLUT**.sup.T and **YLUT**.sup.T representing x and y coordinate data of **XLUT** and **YLUT**, respectively, based on an image transposition or change in orientation.

DETDESC:

DETD(150)

In . . . of a three-dimensional geometric model which may be supplied to terminal 13 from a computer, as discussed with reference to **XLUT** and **YLUT** in FIG. 2.

DETDESC:

DETD(151)

As . . . z value terminals 12, 13 and 14, respectively, operates on ZLUT in the same way shear resampler 22 operates on **YLUT** in FIG. 2, for developing at first and second outputs 83 and 84 scaled x and z coordinate data representing **XLUT** and ZLUT values. It should be noted that as an alternative to developing these **XLUT** values in channel 80, the **XLUT** values developed at output 23 of resampler 22, as shown in channel 20 of FIG. 2, can be supplied to coordinate sampler 88 in channel

80. FIG. 4 shows shear resampler 22A which in addition to developing **XLUT** and **YLUT** as previously described with reference to resampler 22, similarly develops ZLUT so that **XLUT** and ZLUT can be coupled to coordinate resampler 88 from resampler 22A, eliminating the need for an additional shear resampler. . .

DETDESC:

DETD(152)

Coordinate . . . This data, indicated as ZLUT.sub.x, represents the scaled ZLUT values from resampler 82 after resampling in resampler 88 responsive to **XLUT** values to give effect to image compression variations occurring along the direction of the first or x coordinate.

DETDESC:

DETD(164)

Gray . . . from both sources. Ties are arbitrarily resolved in favor of I.sub.xy.sup.T. Finally, in FIG. 26(d), the two spatial lookup tables **XLUT** (on left) and **YLUT** that defined the circular warp, are displayed as intensity images, with y increasing top-to-bottom, and x increasing left-to-right. Bright intensity values in the images of **XLUT** and **YLUT** denote high coordinate values. Note that if the input were to remain undistorted **XLUT** and **YLUT** would be ramps. The deviation from the ramp configuration depicts the amount of deformation which the input image undergoes.

DETDESC:

DETD(165)

FIG. 27 demonstrates the effect of undersampling the spatial lookup tables. The checkerboard is again warped into a circle. However, **XLUT** and **YLUT** were supplied at lower resolution. FIG. 27(a) and (b) show I.sub.xy and I.sub.xy.sup.T, respectively, and FIG. 27(c) shows the output. . .

DETDESC:

DETD(166)

FIG. 28(a) illustrates an example of foldover. FIG. 28(b) shows **XLUT** (on left) and **YLUT**. A foldover occurs because **XLUT** is not monotonically increasing from left to right. In FIG. 29 (a) and (b), the foldover regions are shown magnified. . .

DETDESC:

DETD(167)

FIG. . . . shows the result of bending horizontal rows. For the checkerboard, in FIG. 30(a), and Madonna, FIG. 30(b). FIG. 30(c) illustrates **XLUT** (on left) and **YLUT** and FIG. 30(d) shows S at the output of selector 74. As we scan across the rows in left-to-right order,. . .

DETDESC:

DETD(168)

FIG. 31 shows a vortex warp of the checkerboard in FIG. 31(a) and Madonna in FIG. 31(b). **XLUT** (on left) and **YLUT** are shown in FIG. 31(c) and FIG. 31(d) shows S at the output of selector 74 in FIG. 2.

CLAIMS:

CLMS(9)

9. . . .
means for supplying luminance data each pixel of a plurality of pixels in a two-dimensional image;
x value means for supplying **XLUT** data;
y value means for supplying **YLUT** data;
a first channel, for processing luminance data to derive image values representing preliminary values of output image pixels, comprising:
shear resampler means coupled to said x value and y value means for processing **XLUT** data for developing at a first output scaled **XLUT** data representing **XLUT** of a magnification of said output image and having a plurality of pixel values in place of each pixel value of the basic output image, and for processing **XLUT** data for developing at a second output scaled second coordinate data having a plurality of pixel values for each pixel. . . x direction;
coordinate resampler means coupled to said first and second shear resampler outputs for developing at a first output modified **YLUT** data representing said scaled **YLUT** after resampling to give effect to image compression variations along the x direction, and for developing at a second output. . .
transposing means coupled to said input image, x value and y value means for developing transposed input image and transposed **XLUT** and **YLUT** data representative of said input and output images respectively after transposing their coordinates to a second orientation;

a second channel, for. . .

CLAIMS:

CLMS(12)

12. . . .
means for supplying luminance data each pixel of a plurality of pixels in a two-dimensional image;
x value means for supplying **XLUT** data;
y value means for supplying **YLUT** data;
z value means for supplying ZLUT data;
a first channel, for processing luminance data to derive image values representing preliminary values of output image pixels, comprising:
shear resampler means coupled to said x value and y value means for processing **XLUT** data for developing at a first output scaled **XLUT** data representing **XLUT** of a magnification of said output image and having a plurality of pixel values in place of each pixel value of the basic output image, and for similarly processing **XLUT** data for developing at a second output scaled second coordinate data having a plurality of pixel values for each pixel. . . x direction;
coordinate resampler means coupled to said first and second shear resampler outputs for developing at a first output modified **YLUT** data representing said scaled **YLUT** after resampling to give effect to image compression variations along the x direction, and for developing at a second output. . . the direction of said y coordinate;
a first z channel, for processing ZLUT data, comprising:
terminal means for supplying scaled **XLUT** data as developed at said first shear resampler output of said first channel;
z shear resampler means coupled to said x value, y value and z value means for developing scaled **XLUT** data representing **XLUT** of a magnification of said output image and having a plurality of pixel values in place of each pixel value. . .
coupled to said input image, x value, y value and z value means for developing transposed nput image and transposed **XLUT**, **YLUT** and ZLUT data representative of said input and output images respectively after transposing coordinates of said images to a second. . .
=>

```
=> s l1 and weight###

        721039 WEIGHT###
L8           1 L1 AND WEIGHT###
=> d kwic
US PAT NO:     **5,204,944** [IMAGE AVAILABLE]          L8: 1 of 1
```

SUMMARY:

BSUM(19)

The . . . it covers. Thus each position in the accumulator array evaluates ##EQU1## where f.sub.i is the input value, w.sub.i is the **weight** reflecting its coverage of the output pixel, and N is the total number of deposits into the cell. Note that. . .

DETDESC:

DETD(15)

The . . . be mapped onto the output along a single direction, i.e., with no folds. As each input pixel arrives, it is **weighted** by its partial contribution to the current output pixel and integrated into a single-element accumulator. For input pixels that spread. . .

DETDESC:

DETD(21)

The . . . is given in FIG. 7. For clarity the following notation is used: interpolated input values are written within square brackets, **weights** denoting contributions to output pixels are written within an extra level of parentheses, and input intensity values are printed in. . .

DETDESC:

DETD(22)

The . . . interpolation is used to reconstruct the discrete input. When more than one input element contributes to an output pixel, the **weighted** results are integrated in an accumulator to achieve antialiasing. These two cases are both represented in the above equations, as. . .

DETDESC:

DETD(94)

A . . . The computations are straightforward, and serve a secondary function in that the data entries correspond exactly to the information or **weighting** needed for antialiasing in the intensity resample stage. Thus a local distortion measure is obtained at virtually no additional cost. . .

DETDESC:

DETD(166)

FIG. . . . is more apparent along the fold upon the cheek. The intensity drop is due to the antialiasing filtering that correctly **weighted** the pixels with their area coverage along the edge. This can be resolved by integrating partially visible pixels in front-to-back. . .

```
=> s l1 and coefficient#
        142550 COEFFICIENT#
L9           0 L1 AND COEFFICIENT#
=> s l1 and ("f.sub.i" or "w.sub.i")
        639148 "F"
        795787 "SUB"

       1384402 "I"

          3423 "F.SUB.I"
                 ("F"(W)"SUB"(W)"I")
        282850 "W"
        795787 "SUB"

       1384402 "I"

          1963 "W.SUB.I"
                 ("W"(W)"SUB"(W)"I")
L10          1 L1 AND ("F.SUB.I" OR "W.SUB.I")
=> d kwic
US PAT NO:     **5,204,944** [IMAGE AVAILABLE]         L10: 1 of 1
```

SUMMARY:

BSUM(19)

The . . . the relative area of the output pixel that it covers. Thus each position in the accumulator array evaluates ##EQU1## where **f**.**sub**.**i** is the input value, **w**.**sub**.**i** is the weight

reflecting its coverage of the output pixel, and N is the total number of deposits into the. . .
=>

```
        228930 ALPHA
L11           0 L1 AND ALPHA
=> s l1 and transparen##
        178095 TRANSPAREN##
L12           0 L1 AND TRANSPAREN##
=> s l1 and interpolat###
         18459 INTERPOLAT###
L13           1 L1 AND INTERPOLAT###
=> d kwic
US PAT NO:     **5,204,944** [IMAGE AVAILABLE]         L13: 1 of 1
```

SUMMARY:

BSUM(17)

The forward mapping consists of **interpolating** each input pixel into the output image at positions determined by the X and Y mapping functions. Each input pixel. . .

DETDESC:

DETD(15)

The . . . a single-element accumulator. For input pixels that spread out over many output pixels, image reconstruction is currently implemented with linear **interpolation**. In terms of the input and output streams, one of three conditions is possible:

DETDESC:

DETD(18)

(3) . . . output pixel will be completed without entirely consuming the current input pixel. In this case, a new input value is **interpolated** from the neighboring input pixels at the position where the input was no longer consumed. It is used as the. . .

DETDESC:

DETD(21)

The computation of the resampled intensity values is given in FIG. 7. For clarity the following notation is used: **interpolated** input values are written within square brackets, weights denoting contributions to output pixels are written within an extra level of. . .

DETDESC:

DETD(22)

The algorithm demonstrates both image reconstruction and antialiasing. When not positioned at pixel boundaries in the input stream, linear **interpolation** is used to reconstruct the discrete input. When more than one input element contributes to an output pixel, the weighted. . .

DETDESC:

DETD(24)

While . . . errors proportional to the intensity gradient across the interval. The following resampling algorithm exactly computes the area coverage assuming linear **interpolation** between adjacent intensity values.

DETDESC:

DETD(26)

If . . . to the accumulator. The area is obtained by finding midpoint of the current interval, using the gradient to compute the **interpolation** there, and then multiplying by the length of the interval. With linear **interpolation**, this midpoint rule provides the exact area of the region. Since that input pixel is finished, the next one is. . .

DETDESC:

DETD(27)

If . . . and update the current position and fraction of input pixel remaining. Using the gradient information compute the value of the **interpolated** intensity half way to said boundary, and use this to compute the area coverage and obtain the resampled value, which. . .

DETDESC:

DETD(30)

The . . . resampling intensity. However, with minor modifications, it can be used as a coordinate resampling algorithm (which also benefits from the **interpolation** at midpoints, without area resampling). The modified code is:

DETDESC:

DETD(53)

The . . . register with accumulator 205 is referred to as curi and holds the current value of the intensity at the current **interpolation**. It is needed to compute the area under the **interpolation** for intensity resampling, or it is the value if doing coordinate resampling. The input of the register is connected to. . .

DETDESC:

DETD(54)

The n bit register 206, referred to as nexti, holds the image data for the next point in the **interpolation** (the **interpolation** is between the values curi and nexti). Storing is enabled by the micro-code statement GETNEXTI, with inputs connected to the. . .

DETDESC:

DETD(55)

The n bit register 210, referred to as curL, holds the location data for the current point along the **interpolation**. Storing is enabled when either the micro-control command GETNEXTL or TRUNCCURL is issued. The input is from the 2 to. . .

DETDESC:

DETD(56)

The n bit register 212, referred to as nextL, holds the location data for the next point in the **interpolation** (**interpolation** is from location curL to location nextL). Storing is enabled by the micro-code statement GETNEXTL, with the inputs connected to. . .

DETDESC:

DETD(71)

The . . . differs only in the notable absence of antialiasing filtering--the output coordinate values in YLUT.sub.x are computed by point sampling YLUT. **Interpolation** is used to compute values when no input data is supplied at the resampling locations. However, unlike the intensity resampler. . .

DETDESC:

DETD(81)

This . . . densely. If the continuous mapping functions are no longer available, then new values are computed from the sparse samples by **interpolation**. For example, linear **interpolation** can be used to increase the spatial resolution of XLUT and YLUT. The resulting image in FIG. 13 is shown. . .

DETDESC:

DETD(135)

Shear . . . By providing a plurality of pixel values in place of each pixel value of the basic output image, as by **interpolation** of input coordinate samples as discussed above, the spatial resolution of XLUT and YLUT is, in effect, increased to alleviate. . .

=>